AMENDMENTS TO THE CLAIMS

The following listing of claims replaces all prior versions and listings of claims:

Listing of Claims:

1. (Currently Amended) An encoder for use in a nonvolatile counter comprising:

a logic gate having a first input for receiving an up/down signal, a second input for receiving a most-significant bit signal, and an output;

a plus-one block having an input for receiving an inverted input signal and an output; and

a two-to-one multiplexer having a first input for receiving the inverted input signal, a second input coupled to the output of the plus-one block, a control input coupled to the output of the logic gate, and an output for providing an encoded signal.

2. (Original) The encoder of claim 1 in which the first input is set to a logic one for counting up.

3. (Original) The encoder of claim 1 in which the first input is set to a logic zero for counting down.

4. (Original) The encoder of claim 1 in which the logic gate comprises a XNOR gate.

5. (Original) The encoder of claim 1 in which the plus-one block comprises a combinatorial logic circuit.

6. (Original) The encoder of claim 1 in which the encoded signal is equal to the inverted input signal if the control signal is a logic zero.

7. (Original) The encoder of claim 1 in which the encoded signal is equal to the inverted input signal plus one if the control signal is a logic one.

8. (Original) The encoder of claim 1 in which the nonvolatile counter comprises a ferroelectric nonvolatile counter.

9. (Original) The encoder of claim 1 in which the nonvolatile counter comprises an *m*-bit nonvolatile counter.

10. (Original) The encoder of claim 9 in which *m* is equal to 4.

11. (Original) The encoder of claim 9 in which *m* is equal to 5.

12. (Original) The encoder of claim 9 in which *m* is equal to 40.

13. (Original) A method for encoding an input number *n* comprising:
if *n* is even, providing a conventional binary code of (*n*/2); and
if *n* is odd, providing a complementary binary code of ((*n*-1)/2).

14. (Original) A method for encoding an input signal to an output signal comprising switching all of the bits in the output signal to the logical compliment when counting from an even number to an odd number.

15. (Currently Amended) A <u>data encoder incorporating a</u> method for counting up from an input number $n = d_m d_{m-1} d_{m-2} ... d_1 d_0$ comprising:

if n is even, then $n+1 = \overline{d}_m \overline{d}_{m-1} \overline{d}_{m-2} ... \overline{d}_1 \overline{d}_0$; and

if n is odd, then $n+1 = \overline{d}_m \overline{d}_{m-1} \overline{d}_{m-2} ... \overline{d}_1 \overline{d}_0 + 1$.

16. (Currently Amended) The <u>encoder</u> [[method]] of claim 15 further comprising keeping an output count at a maximal number when counting up from the maximal number instead of rolling the output count over to zero.

17. (Currently Amended) The <u>encoder</u> [[method]] of claim 16 in which the maximal number is equal to 1000 when *m* is equal to four.

18. (Currently Amended) A <u>data encoder incorporating</u> a method for counting down from an input number $n = d_m d_{m-1} d_{m-2} ... d_1 d_0$ comprising:

if n is even, then $n-1=\overline{\overline{d}}_m\overline{d}_{m-1}\overline{d}_{m-2}...\overline{d}_1\overline{d}_0+1$; and

if n is odd, then $n-1=\overline{\overline{d}}_m\overline{d}_{m-1}\overline{d}_{m-2}...\overline{d}_1\overline{d}_0$.

19. (Currently Amended) The encoder [[method]] of claim 18 further comprising keeping an output count at zero when counting down from zero, instead of rolling the output count over to a maximal number.

20. (Currently Amended) The encoder [[method]] of claim 18 in which the maximal number is equal to 1000 when m is equal to four.